# Assignment 3-3 Polynomial Division 1

You are given two polynomials $f(x)$ and $g(x)$ with integer coefficients. In this problem you'll have to find out the quotient $q(x)$ and remainder $r(x)$ of $f(x)$ divided by $g(x)$. For the sake of convenience, we define the zero polynomial as $0x^0$.

## Input

The input consists of $t$ ($30 \le t \le 40$) test cases. The first line of the input contains only positive integer $t$. Then $t$ test cases follow. Each test case consists of two lines which give the two polynomials $f(x)$ and $g(x)$. The polynomials are represented by first an integer $d$ ($0 \le d \le 100$) which represents the degree of the polynomial, followed by $d+1$ integers in the range $[-2^{31}, 2^{31} - 1]$ representing the coefficients of the polynomial. The coefficients are in decreasing order of exponent, and the leading coefficient is not 0. You may assume that the degree of $f(x)$ is greater than or equal to the degree of $g(x)$. Moreover, you may also assume that neither $f(x)$ nor $g(x)$ is the zero polynomial.

## Output

For each test case, you are to output exactly two lines containing, respectively, $q(x)$ and $r(x)$, in the same format as the input. You may assume that the coefficients of $q(x)$ and $r(x)$ can be represented by 32-bit integers. Note that if $r(x)$ is the zero polynomial, the output for $r(x)$ should be '0 0'.

## Sample Input

```
2
3 1 -3 4 -5
2 1 0 2
2 1 0 -1
1 1 1
```

## Sample Output

```
1 1 -3
1 2 1
1 -1
0 0
```

## Suggested data structure

In your program, it is suggested that a polynomial $a_n x^n + a_{n-1}x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0$ is represented by the following array:

| $n$ | $n-1$ | $\cdots$ | 2 | 1 | 0 |
|---|---|---|---|---|---|
| $a_n$ | $a_{n-1}$ | $\cdots$ | $a_2$ | $a_1$ | $a_0$ |

That means the coefficient of the $x^k$ term is stored in location $k$ of the array. For example, the polynomial $-x^3 - 2x^2 + 4$ is represented by the following array:

| 3 | 2 | 1 | 0 |
|---|---|---|---|
| −1 | −2 | 0 | 4 |

Note that the coefficients can be 0.

## Suggested part of the program

You are suggested to write the function `division` to complete the following program which solves this problem. It is also suggested that you declare only one array (or vector) in the function `division`, and don't declare global variables (including global arrays) except `arraySize`.

```
// Polynomial division
#include <iostream>
using namespace std;

const int arraySize = 101;

// quotient = dividend / divisor; remainder = dividend % divisor
// provided that dividendDegree >= divisorDegree, and
// neither dividend nor divisor is the zero polynomial
void division( int dividend[], int divisor[], int quotient[], int remainder[],
   int dividendDegree, int divisorDegree, int &quotientDegree, int
&remainderDegree);

int main()
{
   int T;
   cin >> T;
   for( int t = 0; t < T; t++ )
   {
      int dividend[ arraySize ];
      int dividendDegree;
      cin >> dividendDegree;
      for( int i = dividendDegree; i >= 0; i-- )
         cin >> dividend[ i ];

      int divisor[ arraySize ];
      int divisorDegree;
      cin >> divisorDegree;
      for( int i = divisorDegree; i >= 0; i-- )
         cin >> divisor[ i ];

      int quotient[ arraySize ];
      int remainder[ arraySize ];
      int quotientDegree;
      int remainderDegree;
      division( dividend, divisor, quotient, remainder,
         dividendDegree, divisorDegree, quotientDegree, remainderDegree );

      cout << quotientDegree;
      for( int i = quotientDegree; i >= 0; i-- )
         cout << " " << quotient[ i ];
      cout << endl;

      cout << remainderDegree;
      for( int i = remainderDegree; i >= 0; i-- )
```

```
        cout << " " << remainder[ i ];
      cout << endl;
   }
}

// quotient = dividend / divisor; remainder = dividend % divisor
// provided that dividendDegree >= divisorDegree, and
// neither dividend nor divisor is the zero polynomial
void division( int dividend[], int divisor[], int quotient[], int remainder[],
   int dividendDegree, int divisorDegree, int &quotientDegree,
   int &remainderDegree )
{









}
```